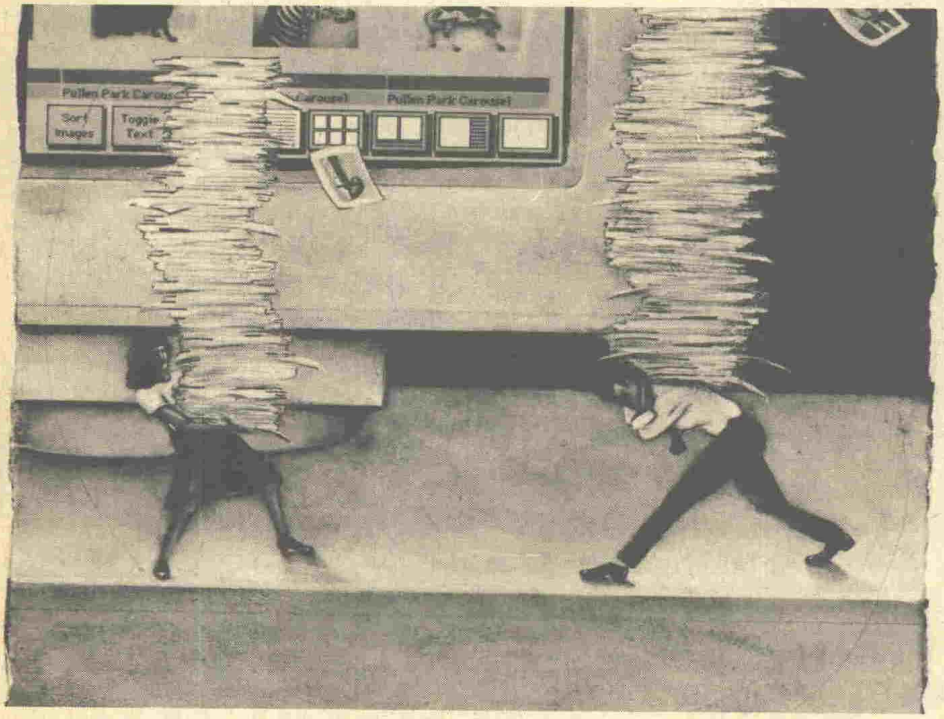
เรื่องน่าอ่าน
ดร.ปัญญา เปรมปรีดิ์

IT Odyssey
การเก็บข้อมูลเพื่อการคำนวณ



เสนอแนวทางการเก็บข้อมูลในหลาย ๆ รูปแบบ ทั้งข้อมูลรวมและข้อมูลกระจายเพื่อใช้ในการคำนวณ

หัวเรื่องอันนี้ผมขอยืมมาจากคุณ Tim Wilson ซึ่งเป็นบรรณาธิการทางเทคนิคของวารสาร Computer World S.E. Asia ฉบับ July 30, 1993 และในขณะเดียวกันก็ขอยืมเอาข้อความส่วนหนึ่งมาให้ท่านผู้อ่านอ่านดู คือคุณทิมได้พูดว่า "And as companies rush to implement distributed computing systems, object oriented technology will unlock the promise of the networks by peving the way for a new class of distributed applications"

ท่านผู้อ่านอ่านแล้วมองเห็นภาพอะไรบ้างไหม ข้อความข้างต้นนี้อ่านแล้วทำให้ตื่นเต้นดี คือมีคำศัพท์ที่กำลังเป็นที่นิยมถึง 3 คำ ซึ่งได้แก่ distributed computing systems, Object oriented technology และ networks หรือ networking 3 คำนี้กำลังได้รับการกล่าวขวัญถึงอยู่ทุกเล่มและทุกเดือน เราพูดกันมากจนทำให้คุณทิมได้ประดิษฐ์ (invent) คำใหม่ขึ้นมา 1 คำ คือ "Distributed Applications" ซึ่งดูแล้วก็น่าฟังดี แต่ผมไม่รู้ว่ามันแปลว่าอะไร ท่านพอจะเดาได้ไหมว่ามันหมายถึงอะไร

ยิ่งไปกว่านั้น เมื่อย้อนกลับไปอ่านซ้ำอีกทีหนึ่งก็จะเห็นว่า เทคนิคออปเจ็กต์โอเรียนเต็ดนั้นไม่ได้เกี่ยวอะไรกับเรื่องของ Distributed Computing และไม่ได้เกี่ยวข้องอะไรกับเน็ตเวอร์กกิง เทคนิคออปเจ็กต์โอเรียนเต็ดนี้ใช้กับเรื่องอะไรก็ได้ การจับเอา 3 คำนี้มาผูกกันจึงอาจก่อให้เกิดความเข้าใจไขว้เขวได้ และคงจะเป็นด้วยเหตุนี้ คุณทิมจึงตั้งชื่อเรื่องว่า "IT Odyssey" เพราะถ้าเชื่อตามท่านก็คงจะหลงไปในจักรวาลจนตาย

ทางเลือกของการคำนวณ

โลกของวิชาชีพ IT (สารสนเทศ) นี้ดูจะสับสนและวุ่นวาย ถ้าใครฟังแล้วไม่หยุดคิดให้ดี ๆ ก็คงจะหลงทิศกันไปได้ วันนี้ผมจะนำเอาเรื่อง Distributed Computing มาพูดกันก่อน

"Distributed Computing" นั้นน่าจะหมายถึงการทำงานที่มีการคำนวณหลาย ๆ แห่ง แต่ทำเพื่อให้เกิดผลงานรวมอันใดอันหนึ่ง เรื่องนี้ดูจะเป็นสิ่งที่หลีกเลี่ยงไม่ได้ ทั้งนี้เพราะการใช้ไมโครคอมพิวเตอร์หลาย ๆ เครื่องนั้นดูจะดีกว่าการใช้เมนเฟรมตัวเดียว เพราะจะประหยัดเงินได้มากกว่า และให้ความคล่องตัวต่อการทำงานมากกว่า เรื่องนี้ผมต้องขอย้ำอยู่บ่อย ๆ ทั้งนี้เพราะคนไทยเราชอบหลงประเด็นอยู่เสมอ ๆ แต่

"Distributed Computing" นี้ไม่ได้บังคับว่าข้อมูลจะต้องกระจายอยู่ตามเครื่องไมโครคอมพิวเตอร์เหล่านี้ มันอาจรวมกันอยู่ที่ศูนย์รวมแห่งใดแห่งหนึ่งก็ได้ และว่ากันที่จริงแล้ว ตัวซีพียูที่จะใช้ก็อาจมีเครื่องเมนเฟรมผสมด้วยก็ได้ นี่ทำให้เรามีหนทางเลือกหลายทาง ตัวอย่างเช่น ใช้เมนเฟรมเป็นตัวเก็บข้อมูลและทำงานคำนวณบางอย่าง แต่งานคำนวณส่วนใหญ่จะอยู่ในไมโครคอมพิวเตอร์ หรือจะใช้ไมโครคอมพิวเตอร์ 1 ตัวเป็นตัวเก็บข้อมูลโดยไม่ต้องคำนวณเลย แล้วใช้ไมโครคอมพิวเตอร์ตัวอื่น ๆ ช่วยการคำนวณก็ได้ หรือจะใช้เป็นเน็ตเวอร์กของไมโครคอมพิวเตอร์และให้กระจายข้อมูลไปอยู่ในเครื่องเหล่านี้ก็ได้ ปัญหาก็คือ ในรูปแบบต่าง ๆ เหล่านี้อันไหนดีที่สุด ซึ่งถ้าท่านไม่คิดให้ดี ๆ ก็อาจพบกับความล้มเหลวได้ เรื่องนี้ขึ้นอยู่กับซิสเต็มส์ซอฟต์แวร์ ถ้าเราไม่มีซิสเต็มส์ซอฟต์แวร์เราก็ทำอะไรไม่ได้ รูปแบบที่ผมเห็นว่าพอจะเป็นไปได้มีดังต่อไปนี้

1. ใช้เมนเฟรมเป็นศูนย์กลาง

ใช้เมนเฟรมเป็นศูนย์กลางแล้วใช้ไมโครคอมพิวเตอร์เป็นเทอร์มินัล แต่ส่วนใหญ่แล้วการคำนวณเข้าไปอยู่ในเมนเฟรม ไมโครคอมพิวเตอร์จะทำตัวเป็นดัมพ์เทอร์มินัลเท่านั้น วิธีนี้จึงเท่ากับเอาไมโครคอมพิวเตอร์เข้ามาแทนที่ดัมพ์เทอร์มินัล ซึ่งจะประหยัดเงินได้ประมาณเทอร์มินัลละ 4 หมื่นบาท (ดัมพ์เทอร์มินัลราคาตัวละ 100,000 บาท ไมโครคอมพิวเตอร์กับอินเตอร์เฟสการ์ดราคาชุดละ 60,000 บาท) งานขนาดกลางที่มีเทอร์มินัล 50 ตัวจะประหยัดเงินได้ประมาณ 2 ล้านบาท (จากราคาเดิมแบบเมนเฟรม + ดัมพ์เทอร์มินัลประมาณ 50 ล้านบาท) เท่านั้น วิธีนี้จึงไม่ใช่วิธีที่ดีอะไรเลย ในปัจจุบันนี้เราก็ทำอย่างนี้กันอยู่แล้ว การที่เราประหยัดได้น้อยกว่าก็เพราะเราไม่ได้กระจายการคำนวณให้มาอยู่ในเครื่องไมโครคอมพิวเตอร์

2. ใช้ไมโครคอมพิวเตอร์เก็บข้อมูล

ใช้ไมโครคอมพิวเตอร์ตัวหนึ่งเป็นตัวเก็บข้อมูล แต่ทำงานคำนวณน้อยมาก ส่วนการคำนวณส่วนใหญ่ให้กระจายไปอยู่ในเครื่องไมโครคอมพิวเตอร์ตัวอื่น ๆ วิธีการนี้ก็คือระบบพีซี-แลนนั่นเอง ซิสเต็มส์ซอฟต์แวร์จะประกอบด้วยโนเวลล์เน็ตแวร์เป็นโอเปอเรตติงซิสเต็มส์ มีบีทรีฟ (Btrieve) เป็นตัวควบคุมจาน-แม่เหล็ก ใช้ดอสและปาสคาลสำหรับสร้างโปรแกรมงาน (Application) วิธีนี้ลดค่าใช้จ่ายจาก 50 ล้านบาทลงมาเหลือแค่ประมาณ 5 ล้านบาท แต่วิธีนี้ท่านต้องเขียนโปรแกรมงานเองเป็นส่วนใหญ่ ซึ่งต้องใช้แรงงานเกือบเท่าเมนเฟรม เช่น งานที่จะใช้แทนเมนเฟรมขนาด 50 เทอร์มินัลนั้นจะต้องเขียนโปรแกรมประมาณ 100-200 โปรแกรม ต้องใช้โปรแกรมเมอร์และแอนาลีสต์ 5-6 คน เป็นเวลา 12-18 เดือน วิธีนี้เป็นวิธีที่ประหยัดเงินได้มากที่สุด

3. ใช้ไมโครคอมพิวเตอร์เก็บข้อมูล แต่ทำงานแบบฐานข้อมูล

ใช้ไมโครคอมพิวเตอร์ตัวหนึ่งเป็นตัวเก็บข้อมูล แต่ทำงานแบบฐานข้อมูล ตัวศูนย์กลางนี้จึงมีการคำนวณอยู่ด้วย ส่วนไมโครคอมพิวเตอร์ที่เป็นเทอร์มินัลนั้นจะทำงานคำนวณเพียงเล็กน้อย เช่น จัดฟอร์มบนหน้าจอภาพ ตรวจสอบลักษณะข้อมูลเพียงเล็ก ๆ น้อย ๆ ระบบแบบนี้เราเรียกว่า "Small Data Base Server" ตัวโอเปอเรตติงซิสเต็มส์ที่ดูแลฐานข้อมูลจะเป็นโอเอสทูของไอบีเอ็ม ตัวระบบฐานข้อมูลเท่าที่เห็นก็มียี่ห้อเดียวคือ ออราเคิล ระบบแบบนี้ใช้กับงานใหญ่ ๆ ไม่ค่อยได้ เท่าที่ทดลองดูจะใช้ได้เพียง 10-15 เทอร์มินัลเท่านั้น เงินลงทุนจะสูงกว่าแบบในข้อ (2) ประมาณ 3-4 เท่า สาเหตุที่แพงขึ้นก็เพราะว่าต้องจ่ายค่าระบบฐานข้อมูลไปเกือบ 1 ล้านบาท แล้วยังต้องขยายเมมโมรีของเทอร์มินัลขึ้นไปเป็น 4 ถึง 8 เมกะไบต์ สำหรับตัวที่วิ่งโอเอสทูนั้นต้องมีเมมโมรีไม่ต่ำกว่า 16 เมกะไบต์ แต่การเขียนโปรแกรมดูจะง่ายขึ้น ทั้งนี้เพราะมีภาษาฐานข้อมูลเข้ามาช่วย แรงงานในการเขียนโปรแกรมจะต่ำกว่าแบบข้อ (2) ประมาณ 30 เปอร์เซ็นต์

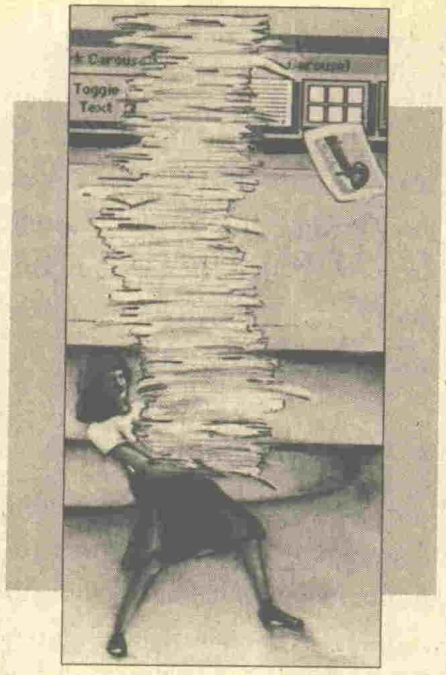
4. ใช้เครื่อง RISC เป็นศูนย์กลาง

ใช้เครื่องประเภท RISC หรือไฮเพาเวอร์เวอร์กสเตชันเป็นศูนย์กลาง เครื่องทุกเครื่องต้องวิ่งภายใต้ยูนิกซ์ แต่เราเก็บข้อมูลไว้ที่ตัวศูนย์กลางเพียงตัวเดียว และต้องมีภาษาฐานข้อมูลควบคุม ภาษาฐานข้อมูลมีให้เลือกมากขึ้น เช่น ออราเคิล อินฟอร์มิกซ์ อินเกรส และซายเบส ตัวเทอร์มินัลต้องเป็นไมโครคอมพิวเตอร์ระดับดี คือรุ่น 80486 ขึ้นไป ต้องมีเมมโมรีไม่ต่ำกว่า 16 เมกะไบต์ ตัวศูนย์กลางต้องมีเมมโมรี 64 ถึง 256 เมกะไบต์ ระบบนี้หลายคนเรียกว่ายูนิกซ์-แลน (UNIX-LAN) แต่จริง ๆ แล้วต้องเรียกว่า "Large Data Base Server System" ระบบแบบนี้ถ้าเอาไปแทนเครื่องเมนเฟรมขนาด 50 เทอร์มินัล จะต้องลงทุนประมาณ 30-40 ล้านบาท หรือคือเป็นราคาใกล้เคียงกับมินิคอมพิวเตอร์ขนาดใหญ่ มันจึงลดค่าใช้จ่ายได้น้อยมาก แต่ก็มีความคล่องตัว

ในการพัฒนาโปรแกรม คือใช้โปรแกรมเมอร์และแอนาลีสต์เหมือนข้อ 3 ระบบตามข้อ 4 นี้จะดีกว่าข้ออื่น ๆ ในแง่ของความเปิดกว้าง

เทคนิคที่ต้องใช้จริง

จากที่กล่าวมาข้างบนนี้จะเห็นว่าไม่มีระบบที่กระจายข้อมูลไปอยู่ในเครื่องที่เป็นเทอร์มินัล จริง ๆ แล้วพอจะทำได้ในกรณีของพีซี-แลน แต่จะช่วยประหยัดได้เพียงเล็กน้อยเท่านั้น สิ่งที่ผมอยากให้ท่านผู้อ่านระมัดระวังเป็นอย่างยิ่งก็คือ มีคนชอบคิดว่า Distributed Computing หมายถึงเน็ตเวอร์กของยูนิกซ์ การเชื่อมโยงเครื่องที่วิ่งภายใต้ยูนิกซ์เข้าหากันนั้นมีมาเกือบ 20 ปีแล้ว และมีขนาดใหญ่มาก เช่น อินเตอร์เน็ตหรือบิตเน็ต พวกนี้มีสมาชิกนับหมื่นเครื่อง แต่เขาเชื่อมโยงกันเพื่อทำอิเล็กทรอนิกส์เมล์ และเพื่อก๊อบปี้ข้อมูลกันเท่านั้น ชาวต่างประเทศเขามีความใฝ่ฝันที่จะให้เชื่อมโยงกันเพื่อ "ทำงานผสานกันเป็นระบบงาน" มานานแล้ว แต่ซิสเต็มส์ซอฟต์แวร์ไม่เหมาะที่จะทำ ตัวอย่างเช่น ระบบฐานข้อมูลที่มีการกระจายข้อมูลอยู่ในเครื่องหลาย ๆ เครื่องนั้นยังไม่มีอันไหนที่ทำงานได้สมบูรณ์ ระบบฐานข้อมูลแบบนี้เรียกว่า "Distributed Data Base" การที่เรายังพัฒนาไม่ได้ก็เพราะเราจะควบคุมการอัพเดตให้ครบถ้วนและถูกต้องได้ยาก ตัวอย่างเช่น ถ้ารายการ (Transaction) นั้นกระทบถึงข้อมูลในเครื่อง 3 เครื่อง เราก็จะต้องได้รับการยืนยัน (Confirmation) จากทุกเครื่องว่าได้อัพเดตเสร็จหมดแล้ว จึงจะแจ้งผลไปยังผู้ทำรายการนั้นได้ แต่เนื่องจากโปรเซสแต่ละอันในเน็ตเวอร์กของยูนิกซ์นั้นสามารถไป



สตาร์ตโปรเซสอื่นได้อีกมากมาย เราจึงไม่รู้ว่าเมื่อไหร่ถึงจะเรียกว่าเป็น "END of TRANSACTION" ได้ การควบคุมเหล่านี้จึงต้องเป็นภาระของโปรแกรมเมอร์เอง ซึ่งเป็นเรื่องที่ยุ่งยากมาก เทคนิคที่เราต้องใช้จริง ๆ นั้นก็คือไคล์เอ็นต์เซอร์ฟเวอร์ หรือคือ ออกแบบระบบให้มีการส่งความต้องการจากโปรแกรมหนึ่งไปยังอีกโปรแกรมหนึ่ง ซึ่งอาจอยู่บนเครื่องเดียวกันหรือคนละเครื่องก็ได้ แล้วให้โปรแกรมที่ 2 ดำเนินการให้ เมื่อเสร็จแล้วก็แจ้งผลกลับไป โปรแกรมที่ 2 นี้อาจสร้างคำสั่งแล้วส่งไปขอให้โปรแกรมที่ 3, 4 ทำให้อีกก็ได้ เมื่อทุกอย่างรายงานผลกลับมาครบถ้วนแล้วจึงจะเรียกว่าเสร็จรายการนั้น เทคนิคนี้กำลังได้รับความสนใจและนำออกปฏิบัติในระดับง่าย ๆ ได้แล้ว ไฟล์เซอร์ฟเวอร์ในระบบพีซี-แลนก็เป็นหนึ่งในผลงานเหล่านี้ แต่อย่างไรก็ตาม การที่จะสร้างระบบงานชนิดที่มีข้อมูลกระจายอยู่ทั่วทั้งเน็ตเวอร์กนั้นยังเป็นเรื่องยากมากในต่างประเทศทำแล้วล้มเหลวมามากแล้ว ดังนั้นคนไทยเราอย่าด่วนสรุปว่า Distributed Computing ภายใต้เน็ตเวอร์กของยูนิกซ์นั้นจะเป็นเรื่องที่จะทำได้ในทันที

บทสรุป

ถ้าจะมองย้อนกลับไปก็จะเห็นว่า Distributed Computing นั้นยังมีข้อจำกัดอยู่ คือ ถ้าเราจะให้กระจายข้อมูลไปอยู่ในเครื่องหลาย ๆ เครื่อง เราจะควบคุมได้ยากมาก เรายังไม่มีซิสเต็มส์ซอฟต์แวร์ที่จะช่วยงานเราได้อย่างแท้จริง ดังนั้นส่วนใหญ่แล้วเรามักจะเอาข้อมูลมารวมไว้ที่เครื่องใดเครื่องหนึ่ง แล้วทำโปรแกรมที่จะควบคุมฐานข้อมูลอันนั้นให้ดี โปรแกรมนี้จะทำหน้าที่เป็นเซอร์ฟเวอร์ ส่วนการคำนวณต่าง ๆ นั้น เราต้องพยายามกระจายออกไปอยู่บนเทอร์มินัลให้มากที่สุดเท่าที่จะมากได้ โปรแกรมหรือโปรเซสที่อยู่ตามเทอร์มินัลต่าง ๆ เหล่านี้จะทำหน้าที่เป็นไคล์เอ็นต์ มันจะยิงคำสั่งเข้ามาให้เซอร์ฟเวอร์ช่วยจัดการกับฐานข้อมูลให้เรา เครื่องที่จะใช้เป็นเทอร์มินัลเหล่านี้ควรเป็นเครื่องไมโครคอมพิวเตอร์ชนิดที่ถูกที่สุด เช่น รุ่น 80386SX วิ่งภายใต้ดอสและมีเมมโมรีเพียง 1-2 เมกะไบต์ก็พอ มันมีพลัง (Processing Power) มากพอที่จะทำงานคำนวณทุกอย่างได้ ระบบแบบนี้ก็คือพีซี-แลนนั่นเอง

เพราะฉะนั้น Distributed Computing = พีซี-แลน # ยูนิกซ์-แลน